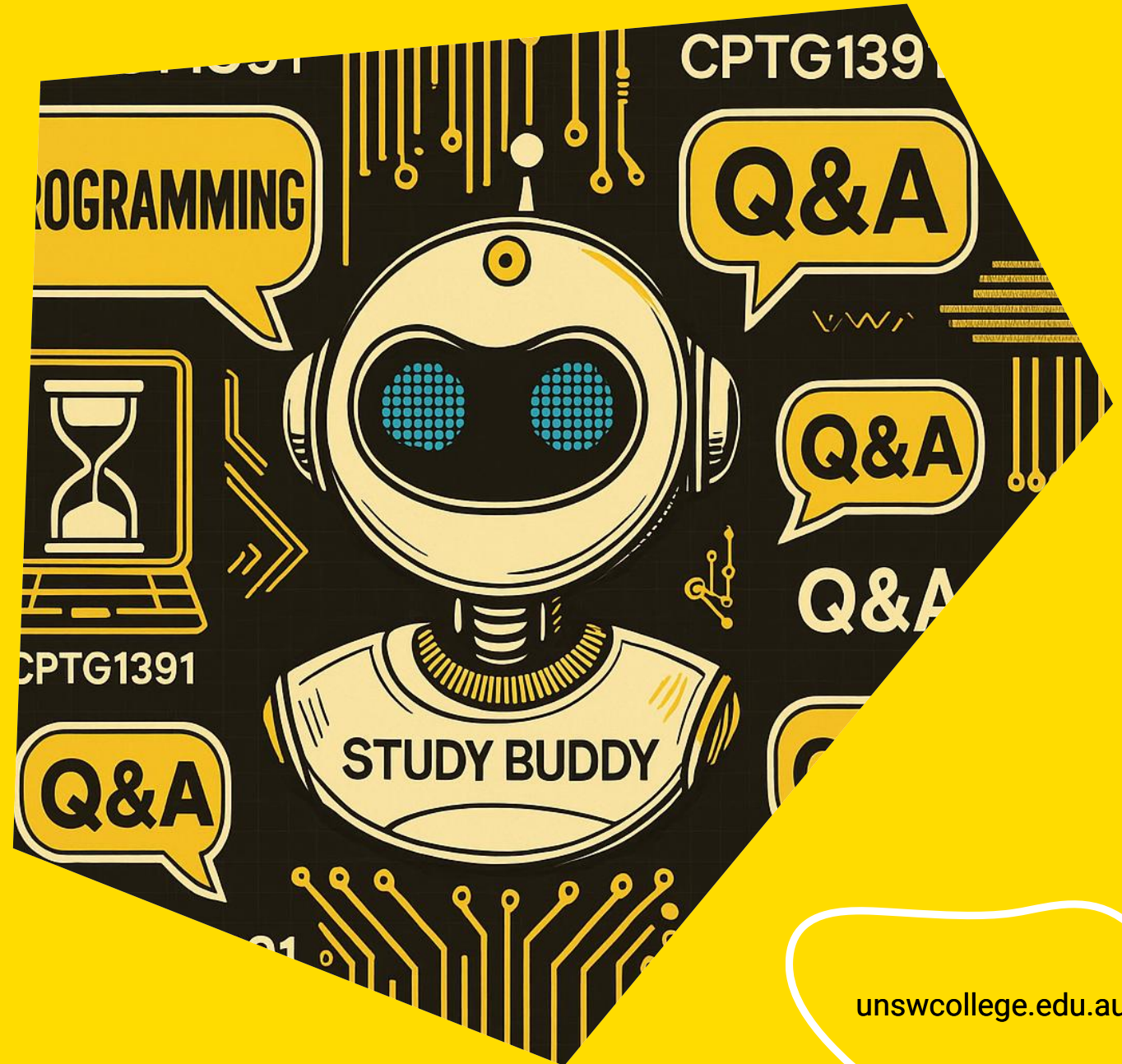


DPST1091 / CPTG1391
Introduction to Programming
Week 1 – Lecture 1

Lecturer and Course Convener:

Dr Pantea Aria

Course Overview



unswcollege.edu.au

Agenda

- **Today**
 - **Course Overview**
 - **Introduction to C**

Who am I?

Dog lover



← Oreo

Coffee lover



Pantea: P.Aria@unswcollege.edu.au

Our Amazing Teachers



Important Resources

The Course page: [Course Website](#)

We use [moodle](#) only to access:

→ Lecture Recordings

→ Link to Online Consultations

**Getting
help**

The Forum

Microsoft Teams

Study Club

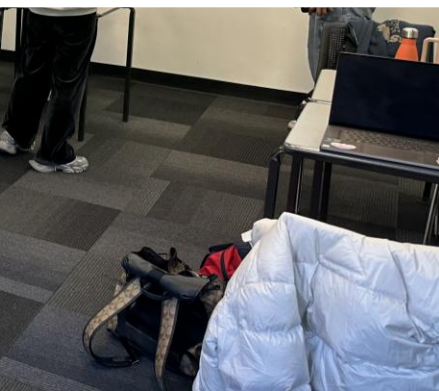
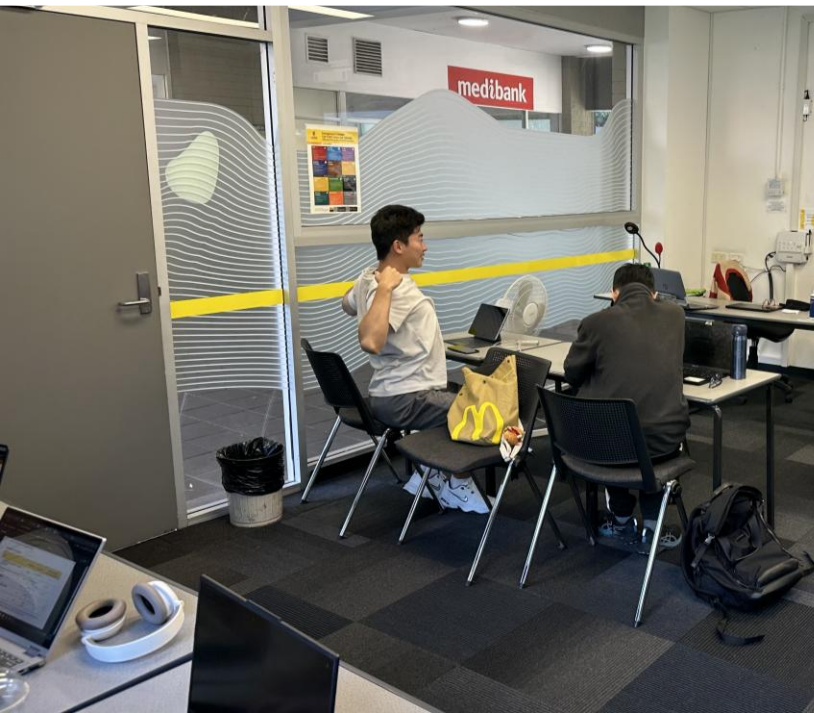
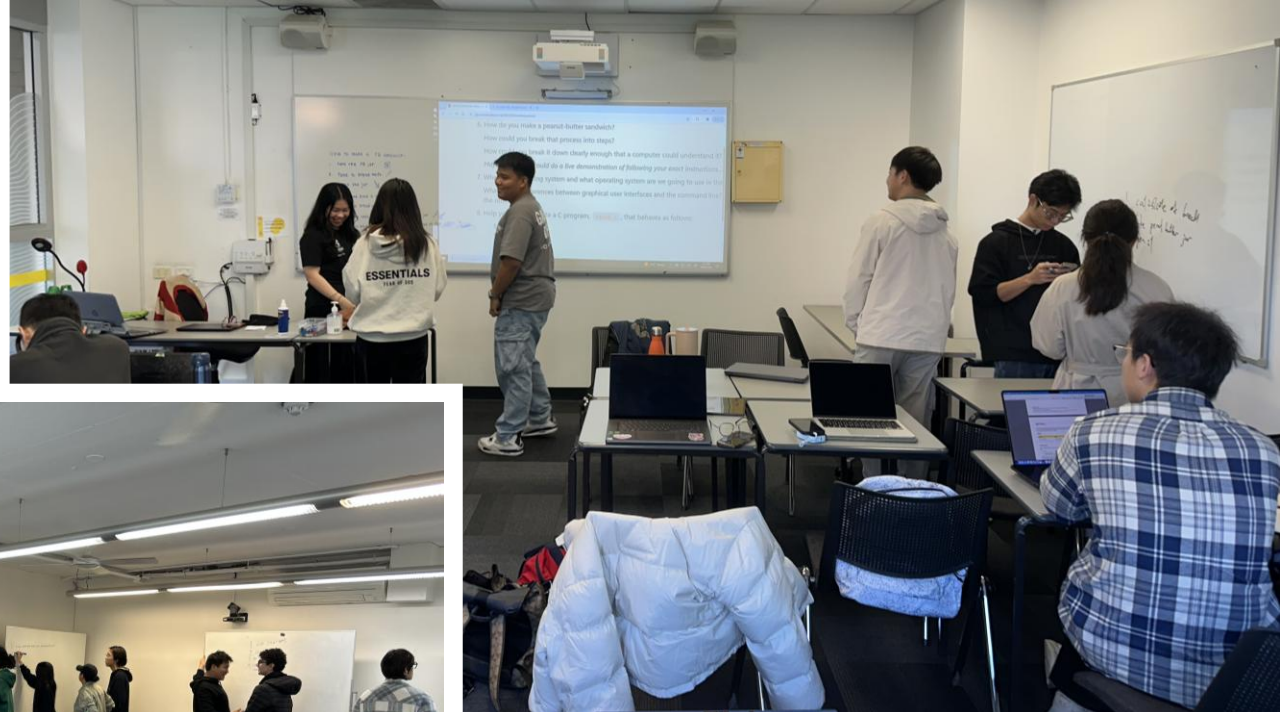
Consultations

Course Format:

- **Weekly lectures**
- **Weekly Tutorials**
- **Weekly Labs**
- **2x Major Assignments**
- **1x Final Exam**

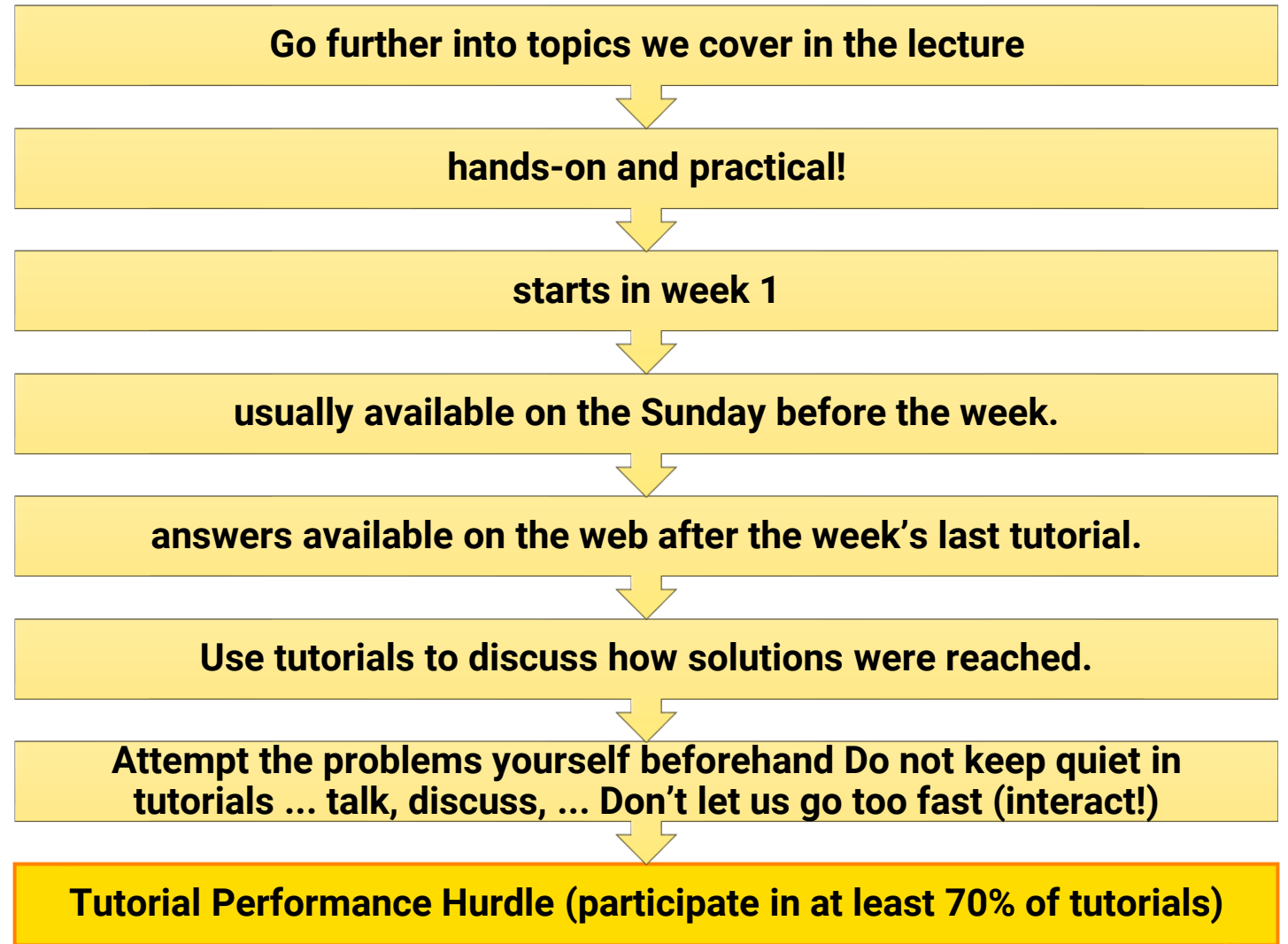
Lectures:

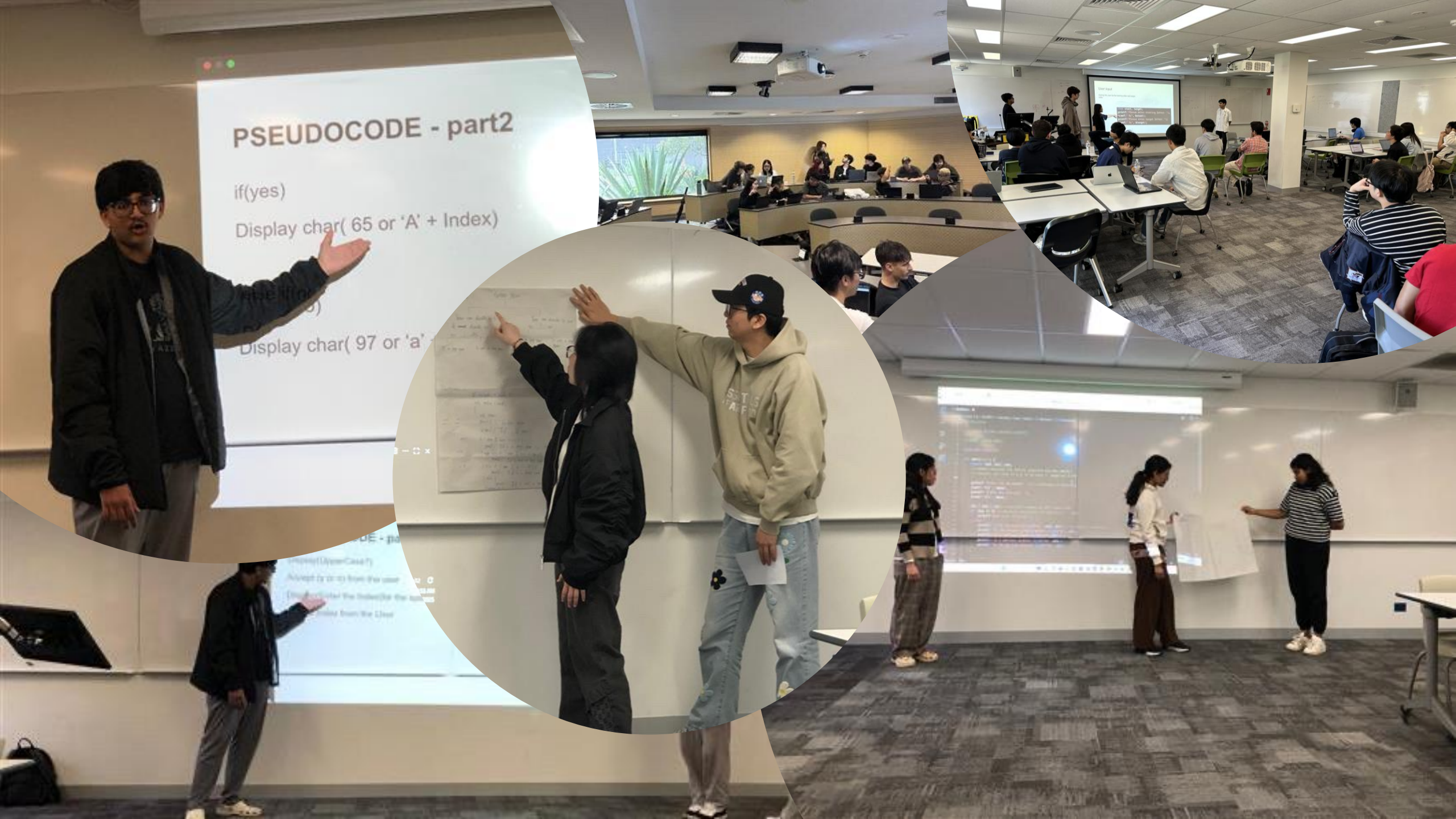
Day	Start Time	End Time	Staff	Location
Mon	6:00 PM	8:00 PM	Pantea	K-F17-LG3 - Rex Vowels Theatre
Wed	9:00 AM	11:00 AM	Pantea	K-J14-G5 - Keith Burrows Theatre



Tutorials:

1 x 2 hours





PSEUDOCODE - part2

if(yes)

Display char(65 or 'A' + Index)

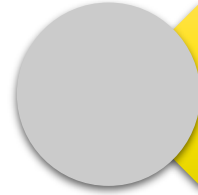
Display char(97 or 'a' + Index)

Handwritten notes on a whiteboard, including pseudocode and diagrams.

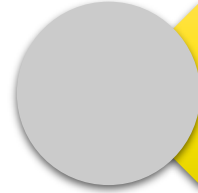
Handwritten notes on a whiteboard, including pseudocode and diagrams.

Labs:

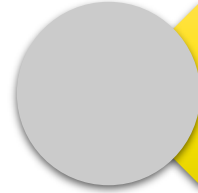
1 x 2 hours



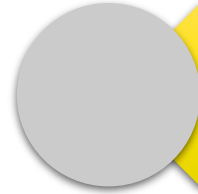
Lab exercise build skills need for assignments & exam



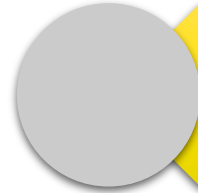
Automarked (with partial marks)



short group **presentation**
(2 to 5 minutes) near the end of the session



15% of final mark.
(30% submissions, 70% presentation)



Full marks possible without completing any challenge exercises

Assignments:

Larger programming problems than the lab exercises

Assignments will be carried out individually

They always take longer than you expect.

Don't leave them to the last minute

Don't plagiarize, we'll get you

- Assignment 1 - 20% (Week 4)
- Assignment 2 - 25% (Week 8)

Code of Conduct:

We offer inclusive learning environment for all students.

No racist/sexist/offensive language or images

No sexually inappropriate behaviour

No bullying, harassing or aggressive behaviors, invasion of privacy

Show respect to your fellow students and the course staff

Plagiarism:

What is plagiarism?

Presenting the (thoughts or) work of another as your own. Cheating of any kind constitutes academic misconduct and carries a range of penalties.

Please read **course outline** for details.

Examples of inappropriate conduct:

- *groupwork on individual assignments (discussion OK)*
- *allowing another student to copy your work*
- *getting your hacker cousin to code for you*
- *purchasing a solution to the assignment*

Remember, you are only cheating yourself and chances are you will get caught!

Plagiarism:

- Lab's submissions must be entirely your **own work**.
- Assignments **must be** entirely your own work.
- You **can not** work on assignment **as a pair** (or group).
- You **can not** use any AI tools for your assignments. (see course outline)
- **Plagiarism will be checked for and penalised.**

How to pass the course:

Hurdle Requirements

To pass the course, you must do all of these:

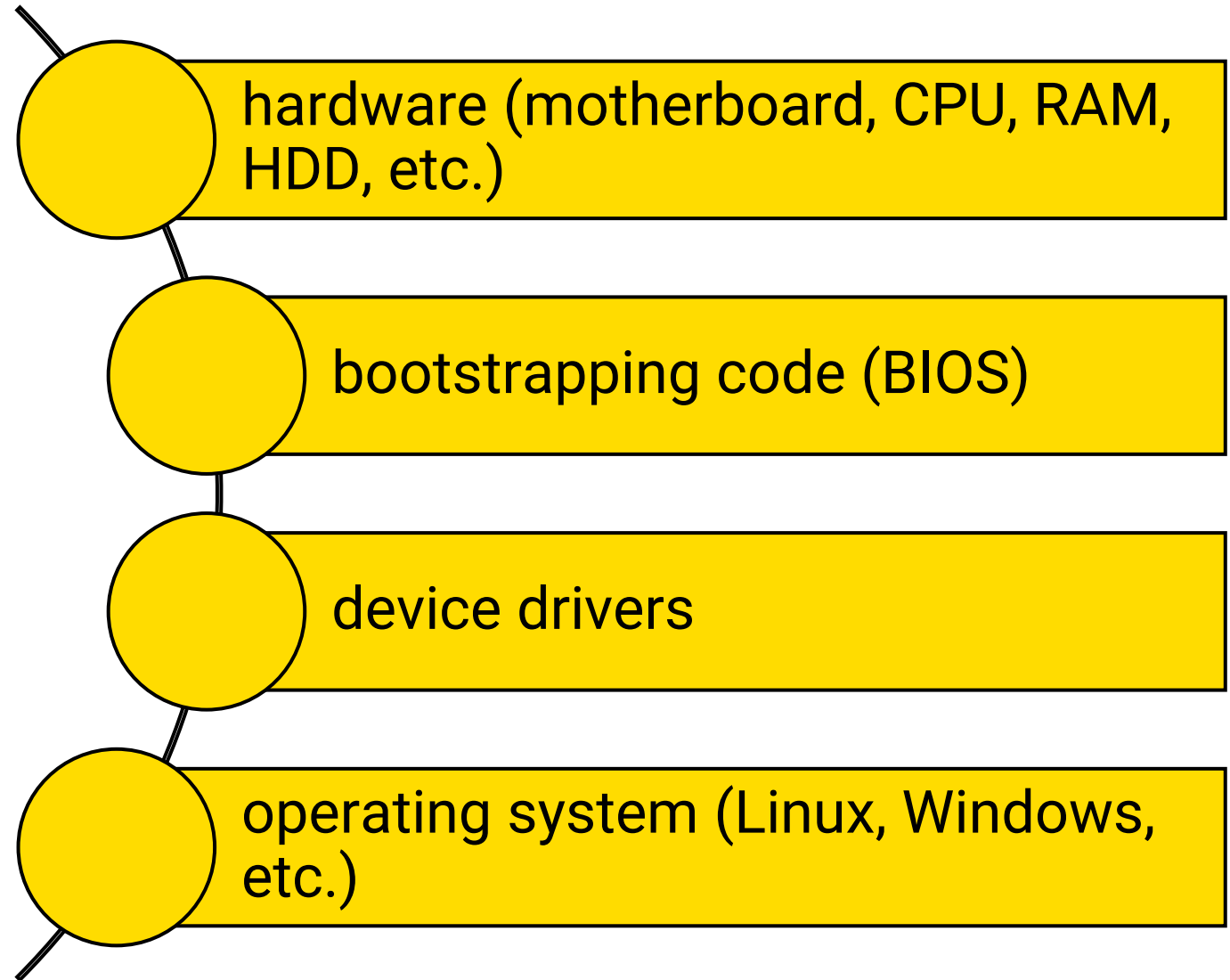
- **Score 50/100 overall**
- **Solve problem using arrays in final exam**
- **Solve problem using linked-lists in final exam**
- **Pass Tutorial Performance Hurdle**

Introduction to Programming

What is a Computer?



Modern computers:



What is Programming?

Programming is the process of writing instructions that tell a computer what to do.

What is an Operating System?

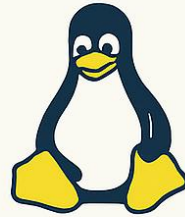
An Operating System is the interface between the user and the computer hardware.

FUNCTION	DESCRIPTION
Program Execution	Runs user applications and system processes
Resource Management	Allocates CPU, memory, and I/O devices efficiently
Security	Protects data and enforces access controls
User Interface	Provides GUI or CLI for user interaction
File System Management	Organizes and stores data on disks

History and Examples

OPERATING SYSTEM	ORIGIN	KEY FEATURES
Unix	1969, AT&T Bell Labs	Multi-user, multitasking, portability
Linux	1991, Linus Torvalds	Open-source, modular, widely used in servers
Windows	1985, Microsoft	GUI-based, broad hardware support
macOS	2001, Apple	Unix-based, sleek GUI, hardware integration
Android	2008, Google	Linux kernel, mobile-focused, app ecosystem

Linux



Using Linux on CSE Machines



Linux Accounts & Access

- Linux is a multi-user operating system
- You will have your own CSE account (username & password)
- Logging in gives you access to your personal files and settings



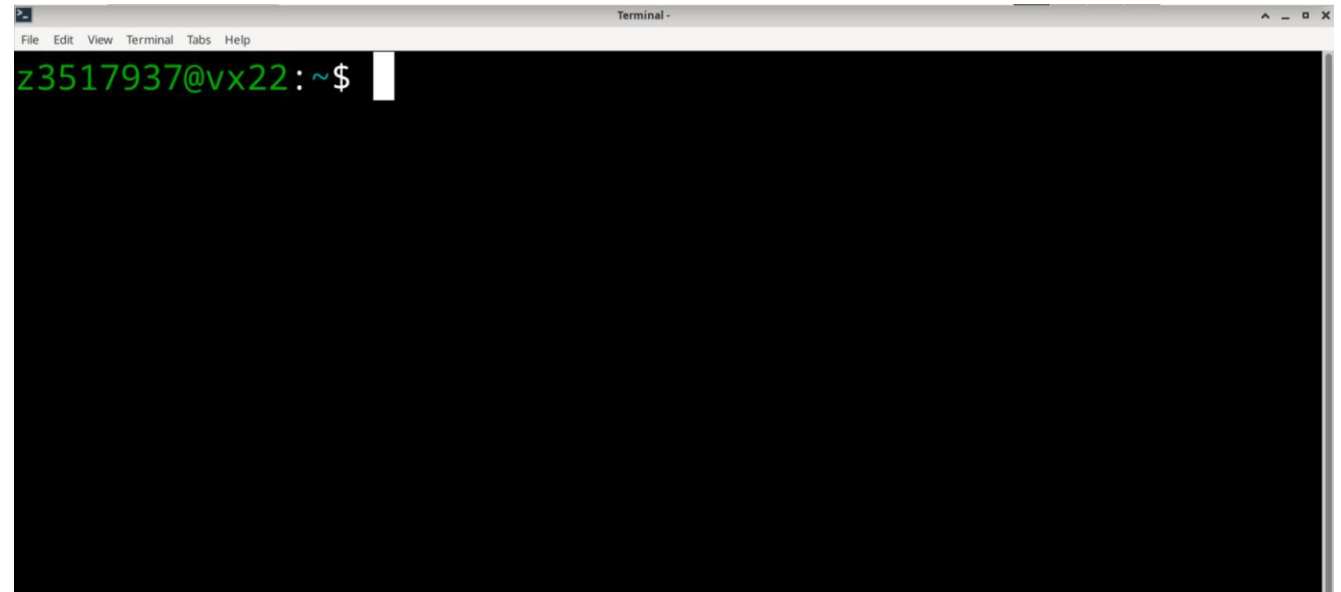
Working in Linux

- Logging in provides access to a terminal window
- A terminal is for text commands executed by the OS.
- Common commands: *ls*, *cd*, *mkdir*, etc.

How do programmers interact with a computer?

The Terminal

- Send text-based commands to our shell (**bash**, **zsh**)
- The **shell** is a program that executes commands and has its own syntax. It returns output which the terminal can display
- Terminal handles user input, rendering shell output



The Prompt

A **prompt** is the line of text displayed by the shell that indicates it is **ready to accept a command**.

```
z3517937@vx22:~$
```

What can I do with the prompt?

Important Linux Commands:

Command	What it does
<i>ls</i>	Lists files and directories
<i>pwd</i>	Shows the current directory path
<i>cd</i>	Changes the current directory
<i>mkdir</i>	Creates a new directory
<i>rmdir</i>	Removes an empty directory
<i>cp</i>	Copies files or directories
<i>mv</i>	Moves or renames files or directories
<i>rm</i>	Deletes files or directories

How to work with Linux from your computer?

https://cgi.cse.unsw.edu.au/~dp1091/26T1/home_computing/vlab.html

Connecting to CSE using VLAB

Accessing the CSE Lab Environment

VLAB stands for *virtual lab*. It is a simple way to access the CSE lab environment from your own computer. You log in using your zID and zPass.

Software you Need

- Download and run the appropriate version of TigerVNC **vncviewer** from the following links:

TigerVNC for Windows (32-bit)

TigerVNC for Windows (64-bit)

TigerVNC for macOS

Move the **VNC viewer** application to somewhere you will be able to find it later. In **macOS**, you may want to move this to your Applications folder. On windows, you may want to move it to your desktop.

Steps to Connect

Every time you want to use VLAB you will need to do the following:

1. Start your TigerVNC **viewer**
 - **On Microsoft Windows**, double click on the exe file.
 - **On macOS**: double click on the dmg file you downloaded and then double click on the viewer icon in the window which opens up.

**How can we ask
computer to solve a
problem for us?**

**For example, how
can I ask computer
to add two
numbers for me?**

Computers language is not easy for us to understand.

For example, to add 5 and 6 and get 11

```
10111000 00000101 00000000 00000000 00000000
10111011 00000110 00000000 00000000 00000000
00000001 11011000
```

Computers need precision!

Programming Languages

They let humans talk to computers

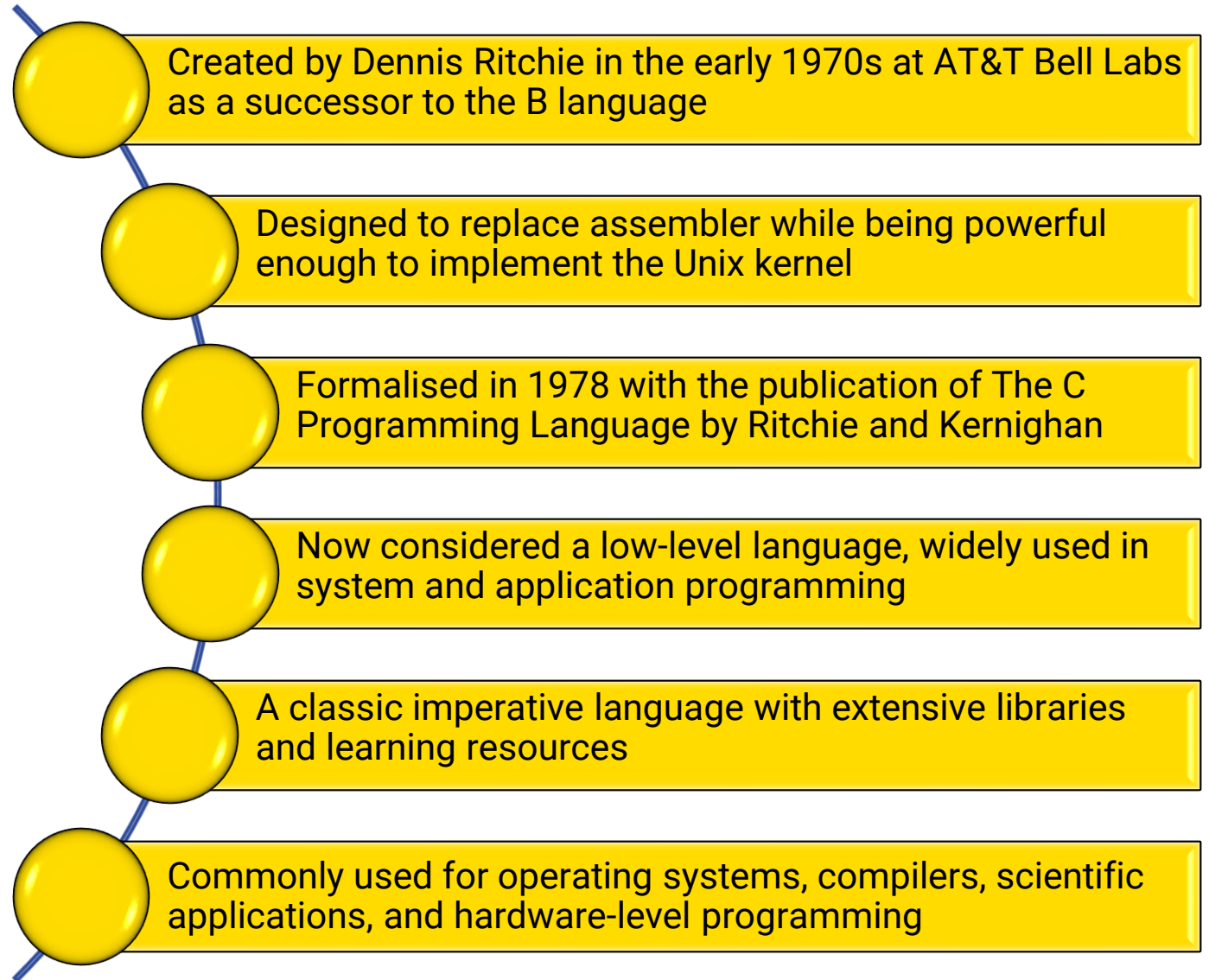
They make complex tasks manageable

They improve speed and accuracy

They support portability

They help solve real-world problems

The C Programming Language



Your first program in C

```
// lecture 1  
  
// your first program in C  
  
// written by: Pantea Aria  
  
#include <stdio.h>  
  
int main(void) {  
    printf("Hello World!\n");  
    return 0;  
}
```

Demo

- Create a .c file using the Terminal
- Write hello world program
- Save it
- Run it

CSE+VLAB - TigerVNC

Applications hello.c (~) - gedit Terminal -

File Edit View Terminal Tabs Help

z3517937@vx22:~\$ gedit hello.c&

Open hello.c Save

1 Start typing your C program here ...

C Tab Width: 4 Ln 1, Col 37 INS

Let's break it down

```
// comments
// lecture 1
// your first program in C
// written by: Pantea Aria

// Include the standard input/output library
for printf
#include <stdio.h>

// Main function: the starting point of the
program
int main(void) {
    // Print "Hello World!" followed by a new
line to the console
    printf("Hello World!\n");

    // Return 0 to indicate the program
finished successfully
    return 0;
}
```

Let's break it down

```
#include <stdio.h>
```

Includes the standard library needed for input and output functions like printf.

```
int main(void) {
```

Defines the main function where the program starts executing.

```
printf("Hello World!\n");
```

Prints the text "Hello World!" followed by a new line to the console.

```
return 0;
```

ends the program and signals that it ran successfully.

```
}
```

Closes the main function.

C Compiler

It takes your **human-readable C program** and **converts** it into a form the **computer** can actually run.

```
z3517937@vx22:~$ gcc hello.c -o hello
z3517937@vx22:~$ ./hello
Hello World!
```

This compiles your hello.c program into an **executable** called hello, and **runs** it

What did we learn today?

Category	Details
Course Focus	Introduction to programming using C , focusing on problem-solving and understanding how computers work
Key Topics	Program structure • Control flow • Functions • Arrays & strings • Pointers • Linked lists
Course Structure	Lectures • Tutorials (2h) • Labs (2h) • 2 Assignments • Final Exam
Assessment	Labs (15%) • A1 (20%) • A2 (25%) • Final Exam
Hurdles	≥50% overall • Arrays + Linked Lists in exam • Tutorial participation ≥70%
Learning Tools	Course website • Moodle (recordings) • Teams • Forum • Consultations
Technical Skills	Linux basics • Terminal commands • Compile & run C programs
Academic Integrity	Individual work only • No AI/tools for assignments • No plagiarism

Voice of the Student

Anonymous ongoing feedback
Anything you wanted to share with me



Voice of the Student



[Voice of the Student – Fill out form](#)

See you soon ...